## AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A two-phase ~~data transfer protocol circuit for a micropipeline, said circuit~~ asynchronous pipeline comprising:

a data path comprising a plurality of processing stages for sequentially processing data, the processing stages interleaved with a plurality of latching stages for holding and propagating the data between the processing stages;

a control ~~element for~~ path for generating ~~micropipeline~~ asynchronous data transfer control signals according to a multiple phase protocol, said control path comprising a plurality of sequentially coupled control elements; and

a pulse generator ~~connected to said micropipeline~~ interface coupled between said control path and said data path and operable to ~~produce pulse signals responsive to~~ translate both rising and falling edges of ~~said~~ the data transfer control signals into data transfer pulses applied to corresponding said latching stages.

2. (Currently Amended) The ~~circuit~~ asynchronous pipeline of claim 1, ~~further comprising a~~ wherein said latching stages comprise level-sensitive latches for holding and propagating data through said ~~micropipeline~~ data path.

3. (Currently Amended) The ~~circuit~~ asynchronous pipeline of claim ~~2~~ 1, wherein said pulse generator ~~is a~~ interface comprises dual-pulse generator means that delivers ~~a~~ data transfer pulses to said ~~level-sensitive~~ latching stages in response to both said rising ~~edge~~ and said falling edges of said data transfer control signals.

4. (Currently Amended) The ~~circuit~~ asynchronous pipeline of claim 1, wherein said control elements ~~is a~~ comprise Muller C-elements.

5. (Currently Amended) The ~~circuit~~ asynchronous pipeline of claim ~~1~~ 3, wherein said dual-pulse generator means comprises a plurality of dual-pulse generators each comprising:

a logic gate having a first input and a second input, wherein said first input is connected to the output of one of said control elements; and

a delay element connected between the output of said control element and said second input, wherein a pulse is produced at the output of said logic gate in accordance with the delay imparted on said data transfer control signal by said delay element.

6. (Currently Amended) The ~~circuit~~ asynchronous pipeline of claim 5, wherein said logic gate is a XOR gate.

7. (Currently Amended) The ~~circuit~~ asynchronous pipeline of claim 5, wherein said delay element comprises an even number of inverters.

8. (Currently Amended) A ~~micropipeline~~ pipeline control circuit for providing asynchronous two-phase data transfer control for a data path having a plurality of sequential data processing stages interleaved with a plurality of latching stages for holding and propagating data between the processing stages, said pipeline control circuit comprising:

a plurality of sequentially coupled C-elements for providing sequential data transfer control among ~~a~~ the plurality of data processing stages ~~within said micropipeline~~, wherein each of said plurality of sequentially coupled C-elements includes a control output, a first input coupled to the control output of a preceding control element, and a second input coupled to the control output of a subsequent control element;

~~a plurality of latches for holding and propagating data through said plurality of processing stages;~~ and

a plurality of dual-pulse generators ~~for~~ each receiving as input the control output from a corresponding one of said plurality of C-elements, each of said dual-pulse generators translating signal transitions from the control outputs of said C-elements into latch control pulses ~~for~~ applied to said plurality of ~~latches~~ latching stages.

9. (Currently Cancelled)

10. (Currently Cancelled)

11. (Currently Cancelled)

12. (Currently Added) The asynchronous pipeline of claim 1, wherein each of said plurality of control elements includes a control output, a req input coupled to the control output of a preceding control element, and an ack input coupled to the control output of a subsequent control element.

13. (Currently Added) The asynchronous pipeline of claim 12, wherein at least one of said control elements is characterized as changing logic states on its control output responsive only to both req and ack inputs changing state.

14. (Currently Added) The asynchronous pipeline of claim 12, wherein for each of said sequentially coupled control elements said control path further comprises a delay element coupled between the req input and the control output of the preceding control element.

15. (Currently Added) The asynchronous pipeline of claim 5, wherein each of said dual-pulse generators is coupled between a control output of one of said control elements and a corresponding latching stage.

16. (Currently Added) A method for implementing an asynchronous two-phase data transfer protocol between stages in a micropipeline, said method comprising:

sequentially processing data within a data path having a plurality of processing stages, wherein the processing stages are interleaved with a plurality of latching stages for holding and propagating the data between the processing stages;

utilizing a control path comprising sequentially coupled control elements to asynchronously generate data transfer control signals according to a multiple phase protocol;

utilizing a pulse generator interface coupled between said control path and said data path to translate both rising and falling edges of the data transfer control signals into data transfer pulses; and

applying the data transfer pulses to corresponding said latching stages.

17. (Currently Added) The method of claim 16, wherein said control path comprises a plurality of sequentially coupled control elements each having a control output, a req input coupled to the control output of a preceding control element, and an ack input coupled to the control output of a subsequent control element, and wherein at least one of said control elements is characterized as changing logic states on its control output responsive only to both req and ack inputs changing state.